On Planar Cubics Through A Point at A Direction

Jin J. Chou[1] *and Matthew W. Blake*[2]

Report RND-92-018 November 1992

NAS Systems Development Branch
NAS Systems Division
NASA Ames Research Center
Mail Stop 258-5
Moffett Field, CA 94035-1000

[1] Computer Sciences Corporation, NASA Ames Research Center, Moffett Field, CA 94035-1000
[2] NASA Ames Research Center, Moffett Field, CA 94035-1000

On Planar Cubics Through A Point at A Direction

Jin J. Chou *
Computer Sciences Corporation
NASA Ames Research Center
M/S T045-2
Moffett Field, CA 94035-1000

Matthew W. Blake
NASA Ames Research Center
M/S T045-2
Moffett Field, CA 94035-5000

Abstract

Cubic planar curves are used frequently to fit planar lists of points. Some authors have suggested methods to construct these curves to utilize the tangent information at the points. The heart of these methods is to find the planar cubic going through three points and the associated tangent-directions. We show that the planar cubics can be found by solving a cubic equation. The result is combined with a previous scheme to produce a better fitting method.

1 Introduction

A parametric curve is a map from a one-dimensional domain space to a three-dimensional space. A planar curve is a curve that lies on a plane. A cubic

*This work was performed for NASA under Contract NAS 2-12961

curve in this paper denotes a polynomial curve or a rational polynomial curve (maybe piecewise continuous) whose degree is cubic.

During geometric processing with computers, at times it is desirable to fit parametric curves through a list of points. The points may come from the numerical solution of a set of partial differential equations, from sampling along an implicit or high order curve, or from a surface-surface intersection calculation. The data points are considered to be "exact," and the fitted curves are required to be as close to the data points as possible.

Tangent-directions of the curves at the data points are frequently readily available from the computation process that generated the points. It is desirable to utilize this tangent-direction information to improve the accuracy of the curve fitting for points between the supplied data points.

In order to capture all the characteristics of the curves and to ensure enough accuracy in the computation, oversampling is a common scheme used by the numerical methods creating the data points. For the convenience of subsequent data storage and handling, it is desired that the curves be represented by a concise representation with less data. Traditionally, the data fitting problem has been handled by passing a cubic spline interpolation curve through the data points and, maybe, through the tangent-directions at the same time [1]—[4]. Such direct interpolation methods do not provide any data reduction. One of the goals of our fitting method is to provide data reduction. In particular, it is hoped that by using the tangent-direction information, a higher data reduction ratio can be achieved while maintaining the same accuracy.

In [5], a sequence of data points in a plane is first fitted with a G^1 piecewise cubic curve, with a cubic piece between every two adjacent data points, and then the number of cubic pieces is reduced by merging adjacent pieces. Two pieces are merged together to form a new cubic piece if the new cubic is close enough to both of the original pieces. The new cubic is found to pass through the two non-shared endpoints of the two adjacent cubics, through a third point located on the curve at around the midpoint of the original cubics, and through the tangent-directions at all the three points. An iterative method is used to find the new cubic.

In [6], a sequence of data points in three-space is fitted by a divide-and-conquer strategy. The center of the algorithm is to find the cubic, with known endpoints and end tangent-directions, passing through a third point in space. It is proved in [6] that, under certain conditions, in three-space there exists a unique cubic going through the third point at a parameter value between 0 and 1. However, when all the points and tangents are in a plane (the planar case), there is one less equation than unknowns. To provide the additional equation, in [6] the parameter value at the third point was assumed. However, it seems natural to use the tangent-direction at the third point to provide the additional equation. In this paper, we investigate the problem of utilizing this tangent information. It should be noted that even when the data points are in three-space, with the divide-and-conquer strategy in [6], it is still necessary to handle planar cases if a portion of the data points lies on a plane.

2 Problem Statement

The problem in finding the new cubic in [5] or the planar case problem in [6] can be stated as the following:

> Given three points P_0, P_3, and P, and three unit vectors, t_0, t_1, t_2, one at each of the points, respectively, find the parametric cubic that has P_0 and P_3 as endpoints, t_0 and t_1 as end tangent-directions, goes through P, and has tangent-direction t_2 at P.

An illustration of the variables is in Figure 1.

If we write the cubic in the Bézier form, with P_0 to P_3 as its control points, the problem can be restated as follows: find the inner control points, P_1 and P_2, such that the cubic passes through P with direction t_2, on condition that P_1 and P_2 lie on the half lines given by t_0 and t_1, respectively.

The Bézier curve can be written as:

$$C(u) = P_0(B_0^3(u) + B_1^3(u)) + \alpha t_0 B_1^3(u) + \beta t_1 B_2^3(u) + P_3(B_2^3(u) + B_3^3(u)) \quad (1)$$

where $B_i^3(u) = \binom{3}{i} u^i (1-u)^{3-i}$ are the cubic Bernstein polynomials, and $P_1 = P_0 + \alpha t_0$ and $P_2 = P_3 + \beta t_1$.

The above equation assumes, without loss of generality, that the cubic starts at $u = 0.$ and ends at $u = 1$.

A couple of issues must be addressed before the answer to the problem becomes useful. First, we have to find a computationally efficient way to obtain the cubic. Second, we have to know what kinds of solutions are possible from the equations. The following sections address these issues.

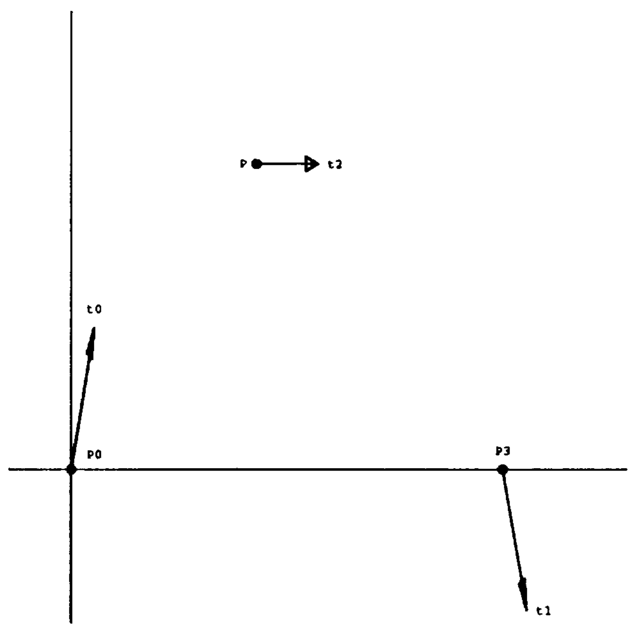

Figure 1: The cubic must go through the three points and tangent-directions.

3 Efficient Computation

In [5], the problem was formulated into two sets of equations. An iterative method was used to solve the equations with the initial guess values of α and β. For each iteration, it is required to solve a closest-point-finding problem for a curve and to solve a 2-by-2 system of linear equations.

We formulate the equations differently and solve the equations directly without iteration. First, the problem statement in Section 2 translates into the following equations for the curve in (1):

$$P = P_0(B_0^3(\bar{u}) + B_1^3(\bar{u})) + \alpha t_0 B_1^3(\bar{u}) + \beta t_1 B_2^3(\bar{u}) + P_3(B_2^3(\bar{u}) + B_3^3(\bar{u})) \quad (2)$$

and

$$\delta t_2 = (P_3 - P_0)B_1^2(\bar{u}) + \alpha t_0(B_0^2(\bar{u}) - B_1^2(\bar{u})) + \beta t_1(B_1^2(\bar{u}) - B_2^2(\bar{u})), \quad (3)$$

where δ is a multiplier, $B_i^2(\bar{u}) = \binom{2}{i}\bar{u}^i(1-\bar{u})^{2-i}$ are the quadratic Bernstein polynomials, and $\bar{u}$ is the parameter at which the curve assumes P.

Without loss of generality, we can assume that the data is in the x-y plane. Then, in the above set of equations, there are four unknowns: α, β, δ, and $\bar{u}$, and four equations: two for the x component and two for the y component. To solve the equations, we define three more vectors:

$$\bar{t}_0 = \hat{k} \times t_0 \quad (4)$$
$$\bar{t}_1 = \hat{k} \times t_1 \quad (5)$$
$$\bar{t}_2 = \hat{k} \times t_2$$

where $\hat{k}$ is the z directional basis vector of unit length. We assume a right-handed Cartesian coordinate system.

Performing dot products on equation (2) with $\bar{t}_0$ and $\bar{t}_1$ and on equation (3) with $\bar{t}_2$ results in the following equations:

$$P \cdot \bar{t}_1 = (P_3 - P_0) \cdot \bar{t}_1(B_0^3(\bar{u}) + B_1^3(\bar{u})) + \alpha t_0 \cdot \bar{t}_1 B_1^3(\bar{u}) + P_0 \cdot \bar{t}_1 \quad (6)$$

$$P \cdot \bar{t}_0 = (P_3 - P_0) \cdot \bar{t}_0(B_0^3(\bar{u}) + B_1^3(\bar{u})) + \beta t_1 \cdot \bar{t}_0 B_2^3(\bar{u}) + P_0 \cdot \bar{t}_0 \quad (7)$$

$$0 = (P_3 - P_0) \cdot \bar{t}_2 B_1^2(\bar{u}) + \alpha t_0 \cdot \bar{t}_2(B_0^2(\bar{u}) - B_1^2(\bar{u})) + \beta t_1 \cdot \bar{t}_2(B_1^2(\bar{u}) - B_2^2(\bar{u})) \quad (8)$$

By substituting α and β from equations (6) and (7) into equation (8) and expanding the Bernstein polynomial out, we can factor $3(1-\bar{u})^2\bar{u}^2$ from the resulting equation and arrive at the following quartic equation.

$$
\begin{aligned}
&((P-P_0)\cdot\bar{t}_1)(t_0\cdot\bar{t}_2) - 2((P-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2) + \\
&(-3((P-P_0)\cdot\bar{t}_1) + 3((P-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2))\bar{u} + \\
&(6((P_3-P_0)\cdot\bar{t}_2)(t_0\cdot\bar{t}_1) - 3((P_3-P_0)\cdot\bar{t}_1)(t_0\cdot\bar{t}_2) + 6((P_3-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2))\bar{u}^2 + \\
&(-12((P_3-P_0)\cdot\bar{t}_2)(t_0\cdot\bar{t}_1) + 11((P_3-P_0)\cdot\bar{t}_1)(t_0\cdot\bar{t}_2) - 13((P_3-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2))\bar{u}^3 + \\
&(6((P_3-P_0)\cdot\bar{t}_2)(t_0\cdot\bar{t}_1) - 6((P_3-P_0)\cdot\bar{t}_1)(t_0\cdot\bar{t}_2) + 6((P_3-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2))\bar{u}^4 = 0
\end{aligned} \tag{9}
$$

Equation (9) can be simplified by applying the following rules, which can be obtained from equation (4) and vector algebra. For any $i, j \in 0, 1, 2$,

$$
\begin{aligned}
t_i \cdot \bar{t}_j &= t_i \cdot (\hat{k} \times t_j) \\
&= -\hat{k}(t_i \times t_j) \\
&= -t_j \cdot \bar{t}_i
\end{aligned} \tag{10}
$$

$$
\begin{aligned}
t_i \times \bar{t}_j &= t_i \times (\hat{k} \times t_j) \\
&= (t_i \cdot t_j)\hat{k} - (t_i \cdot \hat{k})t_j \\
&= (t_i \cdot t_j)\hat{k}
\end{aligned} \tag{11}
$$

and

$$
(a \cdot c)(b \cdot d) - (a \cdot d)(b \cdot c) = (a \times b) \cdot (c \times d) \tag{12}
$$

where a, b, c, and d are vectors, and the term $(t_i \cdot \hat{k})t_j$ in equation (11) is zero since t_i is on the plane.

For example, the coefficient of the quartic term of equation (9) can be simplified to zero:

$$
\begin{aligned}
&(6((P_3-P_0)\cdot\bar{t}_2)(t_0\cdot\bar{t}_1) - 6((P_3-P_0)\cdot\bar{t}_1)(t_0\cdot\bar{t}_2) + 6((P_3-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2)) \\
&= 6[((P_3-P_0)\times\bar{t}_0)\cdot(\bar{t}_2\times\bar{t}_1) + ((P_3-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2)] \\
&= 6[-((P_3-P_0)\cdot\bar{t}_0)\cdot(\bar{t}_2\cdot t_1) + ((P_3-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2)] \\
&= 0
\end{aligned} \tag{13}
$$

After simplification, equation (9) becomes a cubic equation:

$$
\begin{aligned}
&-[((P-P_0)\cdot\bar{t}_2)(t_1\cdot\bar{t}_0) + ((P-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2)] - \\
&3[((P-P_0)\cdot\bar{t}_2)(t_0\cdot\bar{t}_1)]\bar{u} + \\
&3[((P_3-P_0)\cdot\bar{t}_1)(t_0\cdot\bar{t}_2)]\bar{u}^2 + \\
&[((P_3-P_0)\cdot\bar{t}_1)(t_2\cdot\bar{t}_0) + ((P_3-P_0)\cdot\bar{t}_0)(t_2\cdot\bar{t}_1)]\bar{u}^3 = 0
\end{aligned} \tag{14}
$$

4 Solution Discussion

Equation (14) can be solved efficiently and analytically [7]. For a cubic equation with real coefficients, the equation has either one real root or three real roots (with possible repetitive roots). Once $\bar{u}$ is obtained, α, β, and δ can be computed from equations (6), (7), and (3), respectively. Since equation (14) can have three real roots for $\bar{u}$, three sets of solutions $(\bar{u}, \alpha, \beta, \delta)$ are possible. To be a valid solution set, the following conditions have to be met.

$$\begin{aligned} \alpha, \delta &> 0 \\ \beta &< 0 \\ \bar{u} &\in [0,1] \end{aligned} \tag{15}$$

To use the results on curve fitting as done in [5] or [6], either we have to prove that there exists only one valid solution set under some reasonable conditions or we need to handle the other two situations that could occur: none of the solution sets satisfies the conditions, or more than one solution set satisfies the conditions.

Two examples are given to illustrate the possible solution sets. The first example has $P_0 = (0,0), t_0 = (.371391, .928477), P_3 = (7,-1), t_1 = (.351123, -.986329)$, and $P = (3,3)$.

- When $t_2 = (1,0)$, there exists three $\bar{u} \in [0,1]$ but only one valid solution set $(\bar{u} = .391973, \alpha = 5.97567, \beta = -3.53817, \delta = 2.69794)$. Figure 2 shows this curve.
- When $t_2 = (.894427, .447214)$, there is no $\bar{u} \in [0,1]$, hence, there are no valid solutions.
- When $P = (2.5,3), t_2 = (1,0)$, there is one $\bar{u} \in [0,1]$ but there are no valid solutions.

From this example we find that the existence of a valid solution is very sensitive to both the tangent-direction t_2 and the position of P.

The second example has $P_0 = (1,0)$, $t_0 = (0,1)$, $P_3 = (\sqrt{2}/2, \sqrt{2}/2)$, $t_1 = (-\sqrt{2}/2, \sqrt{2}/2)$, $P = (.98465, .174538)$, and $t_2 = (-1.7454, .98465)$. Two sets of valid solutions exist. They are $(\bar{u} = .223347, \alpha = .263335, \beta = -.267153, \delta = .26)$ and $(\bar{u} = .360237, \alpha = .0828732, \beta = -.404803, \delta = .230394)$. This example demonstrates that the solution is not unique: more than one cubic planar curve can pass through the same three points and the same tangent-directions. Figure 3 and Figure 4 show the two valid curves.

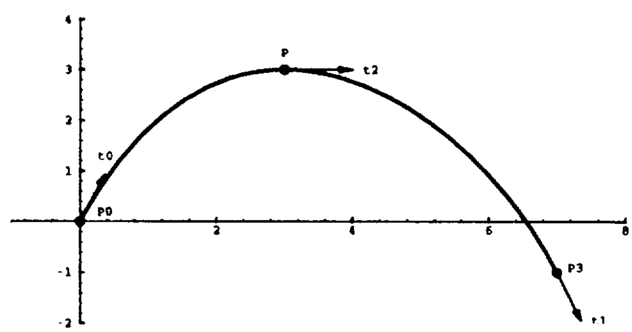

Figure 2: The only valid solution for $t_2 = (1, 0)$.

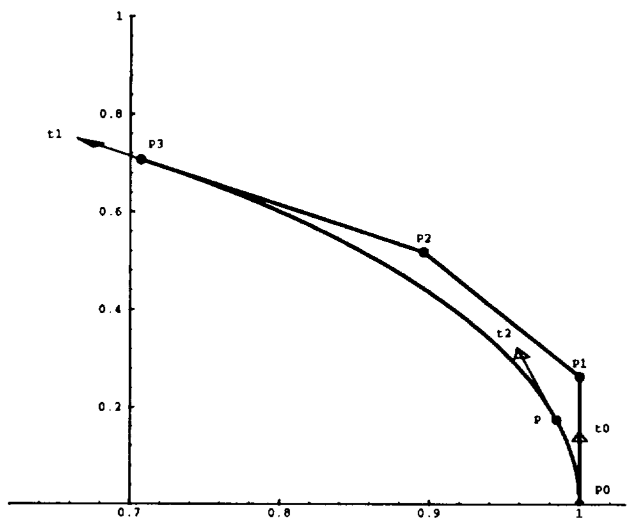

Figure 3: One of the valid solutions with ($\bar{u} = .223347, \alpha = .263335, \beta = -.267153, \delta = .26$).

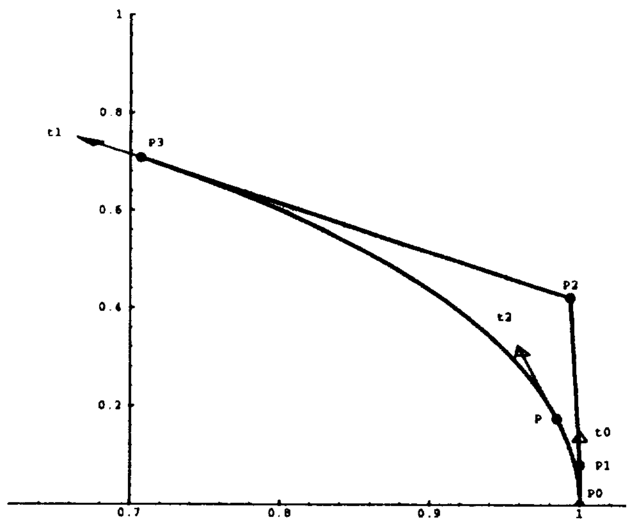

Figure 4: The other valid solution with ($\bar{u} = .360237, \alpha = .0828732, \beta = -.404803, \delta = .230394$).

5 Fitting Examples

In this section, we apply the point interpolation method developed above to curve fitting for a planar set of points with tangent-directions. We follow the method in [6], briefly described below.

In [6] a divide-and-conquer process is applied consistently to divide the data into sections; at the same time, each section is fitted with one rational cubic Bézier curve. After the process is done, all the Bézier curves are connected together to form a single C^1 curve.

During the divide-and-conquer process, given a set of points, if either the control polygon for the fitting curve can not be obtained or the points can not be fitted within a given tolerance, the points are divided into two equal sets, and the fitting process is applied to each of the subsets. In rare cases when only two points remain in the set, a single cubic curve is constructed, using both the position and tangent information at the points.

To find the rational fitting curve for a given set of points, the method in [6] constructs a cubic non-rational Bézier curve first. The control points of the non-rational curve are taken as those for the rational curve. Fitting through the points is done by adjusting the weights of the rational curve.

To obtain the non-rational curve, the cubic interpolating curve through each data point is calculated, one curve per point; the non-rational curve is the average of the interpolating curves. If the interpolating curve for a point can not be found, the curve is excluded from the averaging process. Under this scheme, the fitting process will still proceed properly, even if the interpolating curves for some of the points can not be found. In computing the interpolating curve, a planar case is handled by estimating $\bar{u}$ with a chord length approximation, and the tangent-direction information at the data point is not utilized.

Instead of using an approximation method, we use the cubic non-rational interpolating curve through the data point and the tangent-direction at the point as the interpolating curve. When more than one valid interpolating curve exists, we can either choose an arbitrary one or try to compare the interpolating curves with the one from the neighboring point and choose one that is consistent with the neighboring curve. The latter approach can be done by comparing the αs (and βs) and choosing the one with similar αs (and βs). The new fitting method is exactly the same as what is in [6] except when computing the interpolating curves for planar cases. In the following examples, an arbitrary interpolating curve is used when more than one exists.

Figure 5: A rational fit of 101 points on a half-circle with 10^{-3} tolerance.

We sampled 101 points on a half-circle with unit radius. In Figure 5, these points (shown as dots) and the associated tangent-directions are fitted with a cubic non-uniform B-splines curve with three segments, when the given tolerance is 10^{-3}. In Figure 6, 361 points (not shown) are sampled along an intersection of a plane and a torus. The intersection curve is a quartic algebraic curve. The fitting tolerance is 0.01. The new method is used to fit the spiral data in Figure 10 of [6]. The fitted curve is shown in Figure 7. The points on the top portion of the figure lie on a plane.

Table 1 compares the number of segments in the fitted curves produced by the new method (Method 1) and the method in [6] (Method 2). The data sets in the above examples are used. Note that methods 1 and 2 are the same for non-planar portions of the data.

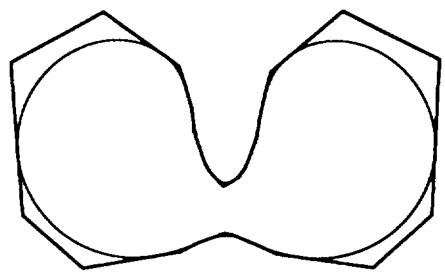

Figure 6: A rational fit of 361 points on the intersection curve of a plane and a torus, with 10^{-2} tolerance.

	Half-Circle		Torus-Plane		Spiral-Data	
Tolerance	10^{-3}	10^{-6}	10^{-2}	10^{-4}	$2 \cdot 10^{-1}$	10^{-3}
No of Segs Method 1	3	19	14	37	5	12
No of Segs Method 2	4	25	15	42	6	18

Table 1: Comparison of the sizes of the fitted curves.

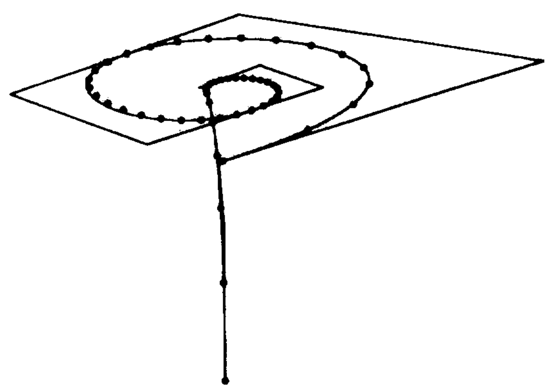

Figure 7: A rational fit of 51 points with $2 \cdot 10^{-1}$ tolerance.

6 Conclusion

We have found a direct and efficient method to solve the problem of finding the planar cubic curves through three points and the associated tangent-directions. We have also proved, by examples, that multiple curves could pass through the same set of data.

We compared this new method with a previous curve fitting method. Examples show some improvement in reducing the sizes of the fitted curves by utilizing the new method, in particular, when the tolerance for fitting is tight.

Our experience, from the above examples, has been that the case in which only one valid solution exists happens most frequently. Moreover, the case in which more than one solution exists occurs much more frequently than the case in which no solution exists.

References

[1] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York (1978).

[2] W. Bohm, G. Farin, and J. Kahmann, A survey of curve and surface methods in CAGD, *Computer Aided Geometric Design*, Vol. 1, No. 1, (1984) pp 1-60.

[3] L. Piegl and W. Tiller, Curve and surface constructions using rational B-splines, *Computer-Aided Design*, Vol. 19, (1987) pp 485-498.

[4] L. Piegl, On NURBS: a survey, *IEEE Computer Graphics & Applications*, Vol. 11, (1991) pp 55-71.

[5] M. Kallay, Approximating a composite cubic curve with one fewer pieces, *Computer-Aided Design*, Vol. 19, (1987) pp 539-543.

[6] J. Chou and L. Piegl, Data Reduction Using Cubic Rational B-splines, *IEEE Computer Graphics & Applications*, Vol. 12, No. 3, (1992) pp 60-68.

[7] W. Beyer, ed., *CRC Handbook of Mathematical Sciences*, CRC Press In., Florida, 1978.